

# COMPLETIONFORMER: DEPTH COMPLETION WITH CONVOLUTIONS AND VISION TRANSFORMERS

**Anonymous authors**

Paper under double-blind review

## ABSTRACT

This paper proposes a joint convolutional attention and Transformer block, which deeply couples the convolutional layer and Vision Transformer into one block, as the basic unit to construct our depth completion model in a pyramidal structure. This hybrid structure naturally benefits both the local connectivity of convolutions and the global context of the Transformer in one single model. As a result, our CompletionFormer outperforms state-of-the-art CNNs-based methods on the outdoor KITTI Depth Completion benchmark and indoor NYUv2 dataset, achieving significantly higher efficiency (near 1/3 FLOPs) compared to pure Transformer-based methods. Especially when the captured depth is highly sparse, the performance gap with other methods gets much larger.

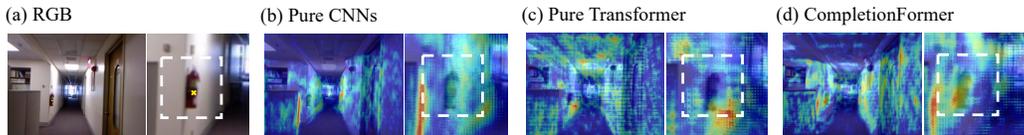


Figure 1: **Comparison of attention maps of pure CNNs, Vision Transformer, and the proposed CompletionFormer with joint CNNs and Transformer structure.** The pixel highlighted with a yellow cross in RGB image (a) is the one we want to observe how the network predicts it. Pure CNNs architecture (b) activates discriminative local regions (*i.e.*, the region on the fire extinguisher), whereas pure Transformer based models (c) activate globally yet fail on local details. In contrast, our full CompletionFormer (d) can retain both the local details and global context.

## 1 INTRODUCTION

Active depth sensing has achieved significant gains in performance and demonstrated its utility in numerous applications, such as autonomous driving and augmented reality. Although depth maps captured by existing commercial depth sensors (*e.g.* Microsoft Kinect Microsoft, Intel RealSense Kesselman et al. (2017)) or depths points within the same scanning line of LiDAR sensors are dense, the distance between valid/correct depth points could still be far owing to the sensor noises, challenging conditions such as transparent, shining, and dark surfaces, or the limited number of scanning lines of LiDAR sensors. To address these issues, depth completion Cheng et al. (2020); Lin et al. (2022); Park et al. (2020); Rho et al. (2022), which targets at completing and reconstructing the whole depth map from sparse depth measurements and a corresponding RGB image (*i.e.* RGBD), has gained much attention in the latest years.

For depth completion, one key point is to get the depth affinity among neighboring pixels so that reliable depth labels can be propagated to the surroundings Cheng et al. (2020; 2018); Hu et al. (2021); Lin et al. (2022); Park et al. (2020). Based on the fact that the given sparse depth could be highly sparse due to noise or even no measurement being returned from the depth sensor, it requires depth completion methods to be capable of 1) detecting depth outliers by measuring the spatial relationship between pixels in both local and global perspectives; 2) fusing valid depth values from close or even extremely far distance points. All these properties ask the network for the potential to capture both local and global correlations between pixels. Current depth completion networks

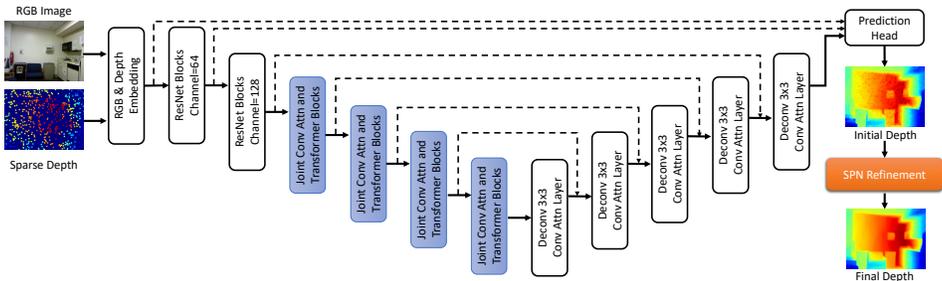


Figure 2: **CompletionFormer Architecture.** Given the sparse depth and corresponding RGB image, a U-Net backbone strengthened with Joint Convolutional Attention and Transformer Block is used to perform the depth and image information interaction at multiple scales. Features from different stages are fused at full resolution and fed for initial prediction. Finally, a spatial propagation network (SPN) is exploited for final refinement.

collect context information with the widely used convolution neural networks (CNNs) Cheng et al. (2020; 2018); Hu et al. (2021); Van Gansbeke et al. (2019); Lin et al. (2022); Qiu et al. (2019); Park et al. (2020); Zhu et al. (2022) or graph neural network Zhao et al. (2021); Xiong et al. (2020). However, both the convolutional layer and graph models can only aggregate within a local region, *e.g.* square kernel in  $3 \times 3$  for convolution and kNN-based neighborhood for graph models Zhao et al. (2021); Xiong et al. (2020), making it still tough to model global long-range relationship, in particular within the shallowest layers of the architecture. Recently, GuideFormer Rho et al. (2022) resorts fully Transformer-based architecture to enable global reasoning. Unfortunately, since Vision Transformers project image patches into vectors through a single step, this causes the loss of local details, resulting in ignoring local feature details in dense prediction tasks Peng et al. (2021); Xu et al. (2021). For depth completion, the limitations affecting pure CNNs or Transformer based networks also manifest, as shown in Fig. 1. Despite *any* distance the reliable depth points could be distributed at, exploring an elegant integration of these two distinct paradigms, *i.e.* CNNs and Transformer, has not been studied for depth completion yet.

In this work, we propose CompletionFormer, a pyramidal architecture coupling CNN-based local features with Transformer-based global representations for enhanced depth completion. Generally, there are two gaps we are facing: 1) the content gap between RGB and depth input; 2) the semantic gap between convolution and Transformer. As for the multimodal input, we propose embedding the RGB and depth information at the early network stage. Thus our CompletionFormer can be implemented in an efficient single-branch architecture as shown in Fig. 2 and multimodal information can be aggregated throughout the whole network. Considering the integration of convolution and Transformer, following the design of CBAM Woo et al. (2018) and Pyramid Vision Transformer Wang et al. (2021), we embrace convolutional attention and Transformer into one block as shown in Fig. 3 and use it as the basic unit to construct our network in multi-scale style. As a result, without any extra module to bridge these gaps Peng et al. (2021); Lee et al. (2022); Rho et al. (2022), every convolution and Transformer layer in the proposed block can access the local and global features. Hence, information exchange and fusion happen at every block of our network.

To summarize, our main contributions are as follows:

- We propose integrating Vision Transformer with convolution layers into one block for depth completion, enabling the network to possess both local and global receptive fields for multimodal information interaction and fusion.
- Taking the proposed joint convolutional attention and Transformer block as the basic unit, we introduce a single-branch network structure, *i.e.* CompletionFormer. This elegant design leads to a comparable computation cost to current CNN-based methods while presenting significantly higher efficiency when compared with pure Transformer based methods.
- Our CompletionFormer yields substantial improvements to depth completion compared to state-of-the-art methods, especially when the provided depth is sparse, as often occurs in practical applications.

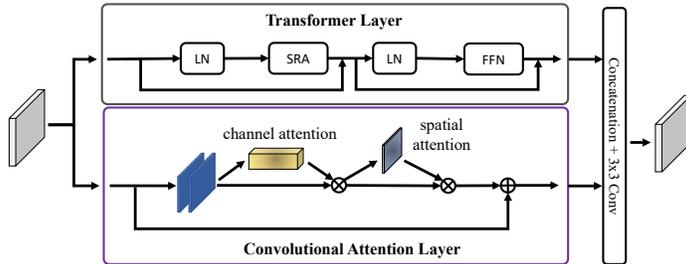


Figure 3: **Joint Convolutional Attention and Transformer Block.** We illustrate the proposed joint convolutional attention and Transformer block which contains two parallel streams, *i.e.* convolutional attention and Transformer layer respectively.

## 2 RELATED WORK

**Depth Completion.** Scene depth completion has become a fundamental task in computer vision with the emergence of active depth sensors. Recently, following the advance of deep learning, fully-convolutional network has been the prototype architecture for current state-of-the-art on depth completion. Ma *et al.* Ma & Karaman (2018); Ma *et al.* (2019) utilize a ResNet He *et al.* (2016) based encoder-decoder architecture, *i.e.* U-Net, within either a supervised or self-supervised framework to predict the dense output. To preserve the accurate measurements in the given sparse depth and also perform refinement over the final depth map, CSPN Cheng *et al.* (2018) appends a convolutional spatial propagation network (SPN Liu *et al.* (2017)) at the end of U-Net to refine its coarse prediction. Based on CSPN, learnable convolutional kernel sizes and a number of iterations are proposed to improve the efficiency Cheng *et al.* (2020), and the performance could be further improved by using unfixed local neighbors Xu *et al.* (2020); Park *et al.* (2020) and independent affinity matrix for each iteration Lin *et al.* (2022). For all these SPN-based methods, while a larger context is observed within recurrent processing, the performance is limited by the capacity of the convolutional U-Net backbone. Accordingly, we strengthen the expressivity of the U-Net backbone with local and global coherent context information, proving effective in improving performance.

Rather than depending on a single branch, multi-branch networks Van Gansbeke *et al.* (2019); Liu *et al.* (2021a); Hu *et al.* (2021); Tang *et al.* (2020); Qiu *et al.* (2019); Nazir *et al.* (2022); Zhang & Funkhouser (2018) are also adopted to perform multi-modal fusion. The common way to fuse the multi-modal information is simple concatenation or element-wise summation operation. More sophisticated strategies like image-guided spatially-variant convolution Tang *et al.* (2020); Yan *et al.* (2022), channel-wise canonical correlation analysis Zhong *et al.* (2019), neighbour attention mechanism Zhang *et al.* (2020) and attention-based graph propagation Zhao *et al.* (2021); Xiong *et al.* (2020) were also proposed to enhance local information interaction and fusion. Instead of pixel-wise operation or local fusion, recently, GuideFormer Rho *et al.* (2022) proposed a dual-branch fully Transformer-based network to embed the RGB and depth input separately, and an extra module is further designed to capture inter-modal dependencies. The independent design for each input source leads to huge computation costs (near 2T FLOPs with the  $352 \times 1216$  input). In contrast, our CompletionFormer in one branch brings significant efficiency (559.5G FLOPs), and the included convolutional attention layer complements the disadvantage of a Transformer in local details.

**Vision Transformer.** Transformers Lee *et al.* (2022); Liu *et al.* (2021b) are first introduced in natural language processing Vaswani *et al.* (2017), then also showing great potential in the fields of image classification Dosovitskiy *et al.* (2020), object detection Liu *et al.* (2021b); Lee *et al.* (2022); Xu *et al.* (2021) and semantic segmentation Xie *et al.* (2021). Tasks related to 3D vision have also benefited from the enriched modeling capability of Transformer, such as stereo matching Li *et al.* (2021; 2022a), supervised Ranftl *et al.* (2021); Li *et al.* (2022b) and unsupervised monocular depth estimation Zhao *et al.* (2022), optical flow Sui *et al.* (2022); Jiang *et al.* (2021) and also depth completion Rho *et al.* (2022). Instead of relying on pure Vision Transformer Rho *et al.* (2022), in this paper, we explore the combination of Transformer and convolution into one block for depth completion. Compared to the general backbone networks (*e.g.* fully CNN-based ResNet He *et al.* (2016), pure Transformer-based Swin Transformer Liu *et al.* (2021b) and MPViT Lee *et al.* (2022) which contains both convolutional and Transformer paths), our proposed joint convolutional attention and

CompletionFormer	#Layers	Params (M)	FLOPs (G)
Tiny	[2, 2, 2, 2]	41.5	191.7
Small	[3, 3, 6, 3]	78.3	231.8
Base	[3, 3, 18, 3]	142.4	301.9

Table 1: **CompletionFormer Configurations.** #Layers denotes the number of our joint blocks in each stage. For all model variants, the channels of 4 stages are 64, 128, 320, 512, respectively. FLOPs are measured using  $480 \times 640$  input image.

Transformer block achieves much higher efficiency and performance on public benchmarks Silberman et al. (2012); Uhrig et al. (2017)

### 3 METHOD

In practical applications, depth maps captured by sensors present various levels of sparsity and noise. Our goal is to introduce both the local features and global context information into the depth completion task so that it can gather reliable depth hints from any distance. The overall diagram of our CompletionFormer is shown in Figure 2. After obtaining depth and RGB image embedding, a backbone constructed by our joint convolutional attention and Transformer block is used for feature extraction at multiple scales. Then, a decoder enhanced by spatial and channel-wise attention provides full-resolution features for initial depth prediction. Finally, we refine the initial estimation with a spatial propagation network.

#### 3.1 RGB AND DEPTH EMBEDDING

For depth completion, multimodal information fusion at an early stage has several advantages, 1) it makes the feature vector of each pixel possess both the RGB and depth information so that pixels with invalid depth still have a chance to be corrected by reliable depth measurements according to appearance similarity; 2) only one branch is required for the following network, which enables much efficient implementation. Therefore, we firstly use two separate convolutions to encode the input sparse depth map  $S$  and RGB image  $I$ . The outputs are concatenated and further processed by another convolution layer to get the raw feature containing contents from both sources.

#### 3.2 JOINT CONVOLUTIONAL ATTENTION AND TRANSFORMER ENCODER

For depth completion, it has been extensively studied how to build connections between pixels to implement depth propagation from reliable pixels while avoiding incorrect ones. Recently, convolution layer Cheng et al. (2020; 2018); Hu et al. (2021); Van Gansbeke et al. (2019); Lin et al. (2022); Qiu et al. (2019); Park et al. (2020); Zhu et al. (2022) or attention based graph propagation Zhao et al. (2021); Xiong et al. (2020) has been the dominant operation for this purpose. Although a fully Transformer-based network Rho et al. (2022) has also been adopted for this purpose, it shows worse results and much higher computational cost compared to pure CNNs-based methods. Considering the complementary properties of these two-style operations, within CompletionFormer, we propose a joint design to take advantage of both the local details of convolutions and the global dependencies of Transformer, while enjoying less computation cost compared to fully CNNs-based or Transformer-based networks as shown in Tab. 2.

Specifically, our Transformer encoder takes five stages, allowing features representation at different scales to communicate with each other effectively. In the first stage, to decrease the computation cost and memory overhead introduced by the Transformer layer, we use a series of BasicBlocks from ResNet34 He et al. (2016) to process and finally get downsampled feature map  $F_1$  at half resolution. For the next four stages, we introduce our proposed joint convolutional attention and Transformer block as the basic unit for framework design.

Basically, for each stage  $i \in \{2, 3, 4, 5\}$ , it consists of a patch embedding module and  $L_i$  repeated joint convolutional attention and Transformer blocks. The patch embedding module firstly divides the feature map  $F_{i-1}$  from previous stage  $i - 1$  into patches with size  $2 \times 2$ . We implement it with a  $3 \times 3$  convolution layer and stride set to 2 as Wang et al. (2021), so it actually halves resolution

for features  $F_{i-1}$  and thus allows for obtaining a features pyramid  $\{F_2, F_3, F_4, F_5\}$ , whose resolutions are  $\{1/4, 1/8, 1/16, 1/32\}$  with respect to the input image. Furthermore, position embedding is also included in the embedded patches and passed through the following joint blocks.

**Joint Convolutional Attention and Transformer Block.** Although Vision Transformer Liu et al. (2021b); Dosovitskiy et al. (2020); Wang et al. (2021) has been proved to be effective in long-range dependencies modelling, it is likely to ignore local relationships even with position embedding in each patch. To cope with the issue raised by irregularly spaced and sparse depth measurements from sensors, we integrate the Transformer layer and a convolutional attention layer into one block for feature extraction, in which the convolutional attention layer is responsible for local details. In overview, our block is organized in a parallel manner as shown in Fig. 3. The Transformer layer is implemented in an efficient way as in Pyramid Vision Transformer Wang et al. (2021), which contains a spatial-reduction attention (SRA) layer with multi-head mechanism and a feed-forward layer (FNN). Given input features  $F \in \mathbb{R}^{H_i \times W_i \times C}$  from the patch embedding module or last joint block (with  $H_i$  and  $W_i$  height and width of features at stage  $i$ , and  $C$  number of channel), we firstly normalize it with layer normalization Ba et al. (2016) (LN) and then flatten it into vector tokens  $X \in \mathbb{R}^{N \times C}$ , where  $N$  is the number of tokens and equals to  $H_i \times W_i$ , *i.e.* the number of all pixels in the  $F$ . Using learned linear transformations  $W^Q$ ,  $W^K$ , and  $W^V \in \mathbb{R}^{C \times C}$ , tokens  $X$  are projected into corresponding query  $Q$ , key  $K$ , and value vectors  $V \in \mathbb{R}^{N \times C}$ . Here, the spatial scale of  $K$  and  $V$  is further reduced to decrease memory consumption, and then self-attention is performed as:

$$\text{Attention}(Q, K, V) = \text{Softmax}\left(\frac{QK^T}{\sqrt{C_{head}}}\right)V, \quad (1)$$

with  $C_{head}$  the channel dimension of each attention head in SRA. According to Eq. (1), each token in the entire input space  $F$  is matched with any tokens, including itself. Our depth completion network benefits from the self-attention mechanism in two folds: 1) it extends the receptive field of our network to the full image in each Transformer layer; 2) as we have embedded each token with both depth and RGB image information, the self-attention mechanism explicitly compare the similarity of each pixel not only by appearance, but also by depth with dot-product operation. Thus, reliable depth information can be broadcasted to the whole image, enabling to correct erroneous pixels.

Inspired by CBAM Woo et al. (2018), we boost the representation power of the convolutional path with channel and spatial attention. On the one hand, it helps to model locally accurate attention and reduce noise. On the other hand, due to the semantic gap between convolution and Transformer, the increased modeling capacity by using the attention mechanism enables this path to focus on important features provided by the Transformer layer while suppressing the unnecessary ones. Finally, by concatenating the reshaped feature from the Transformer-based path, we fuse the two paths with a  $3 \times 3$  convolution and send it to the next block or stage.

Taking the proposed joint block as the basic unit, we build stages 2-5 with repeated configurations. As reported in Tab. 1, we scale-up the 4 stages in CompletionFormer from tiny, small to base scale. Results shown in Tab. 2 demonstrate the superiority of our design compared to recent Vision Transformers Lee et al. (2022); Liu et al. (2021b); Wang et al. (2021) in depth completion task.

### 3.3 DECODER AND MULTISCALE FUSION MODULE

In the decoder, outputs from each encoding layer are concatenated and further processed by the corresponding decoding layers via skip connections. To accommodate diverse scale features better, the features from last decoder layer is upsampled to current scale with a deconvolution layer and convolutional attention mechanism Woo et al. (2018) is also exploited to strengthen feature fusion in channel and spatial dimensions. Finally, the fused result from the decoder is concatenated with features from stage one and fed to the first convolution layer of the prediction head. Its output is concatenated with the raw feature from RGB and depth embedding module (Sec. 3.1) and sent to another convolution, which is in charge of initial depth prediction  $D^0$ .

### 3.4 SPN REFINEMENT AND LOSS FUNCTION

Depth by direct regression from U-Net is still blurry and misaligned with the given image structure. Here, we adopt the non-local spatial propagation network Park et al. (2020) (NLSPN) for further

refinement. Specifically, let  $D^t = (d_{u,v}^t) \in \mathbb{R}^{H \times W}$  denotes the 2D depth map updated by spatial propagation at step  $t$ , where  $d_{u,v}^t$  denotes the depth value at pixel  $(u, v)$ , and  $H, W$  denotes the height and width of the  $D^t$  respectively. The propagation of  $d_{u,v}^t$  at step  $t$  with its non-local neighbors  $N_{u,v}^{NL}$  is defined as follows:

$$d_{u,v}^t = w_{u,v}(0, 0)d_{u,v}^{t-1} + \sum_{(i,j) \in N_{u,v}^{NL}, i \neq 0, j \neq 0} w_{u,v}(i, j)d_{i,j}^{t-1}, \quad (2)$$

where  $w_{u,v}(i, j) \in (-1, 1)$  describes the affinity weight between the reference pixel at  $(u, v)$  and its neighbor pixel at  $(i, j)$ , and  $w_{u,v}(0, 0) = 1 - \sum_{(i,j) \in N_{u,v}, i \neq 0, j \neq 0} w_{u,v}(i, j)$  stands for how much the original depth  $d_{u,v}^{t-1}$  will be preserved. Moreover, the affinity matrix  $w$  is modulated by a predicted confidence map to prevent less confident pixels from propagating into neighbors regardless of how large the affinity is.

Similar to NLSPN Park et al. (2020), features from U-Net backbone are shared for estimating affinity matrix  $w$ , non-local neighbors  $N_{u,v}^{NL}$ , confidence map and initial depth  $D^0$  estimation. Specifically, the prediction is achieved by several separate heads, taking almost the same design used to estimate  $D^0$ , and only the output channel is changed according to the output type. After  $K$  steps ( $K$  is set to 6 in our CompletionFormer) spatial propagation, we get the final refined depth map  $D^K$ .

Finally, following Park et al. (2020), a combined  $L_1$  and  $L_2$  loss is employed to supervise the network training as follows:

$$L(\hat{D}, D^{gt}) = \frac{1}{|V|} \sum_{v \in V} (|\hat{D}_v - D_v^{gt}| + |\hat{D}_v - D_v^{gt}|^2), \quad (3)$$

where  $\hat{D} = D^K$ , and  $V$  is the set of pixels with valid depth in ground truth depth  $D^{gt}$ , and  $|V|$  denotes the size of set  $V$ .

## 4 EXPERIMENTS

### 4.1 DATASETS

**NYUv2 Dataset Silberman et al. (2012):** it consists of RGB and depth images captured by Microsoft Kinect Microsoft in 464 indoor scenes. Following the similar setting of previous depth completion methods Park et al. (2020); Zhu et al. (2022), our method is trained on 50,000 images uniformly sampled from the training set and tested on the 654 images from the official labeled test set for evaluation. For both training and test sets, the original frames of size  $640 \times 480$  are half down-sampled with bilinear interpolation and then center-cropped to  $304 \times 228$ . The sparse input depth is generated by random sampling from the dense ground truth.

**KITTI Depth Completion (DC) Dataset Uhrig et al. (2017):** it contains 86 898 training data, 1 000 selected for validation, and 1 000 for testing without ground truth. The original depth map obtained by the Velodyne HDL-64e is sparse, covering about 5.9% pixels. The dense ground truth is generated by collecting LiDAR scans from 11 consecutive temporal frames into one, producing near 30% annotated pixels. These registered points are verified with the stereo image pairs to eliminate noisy values. Since there is no LiDAR return at the top of the image, following Park et al. (2020), input images are bottom center-cropped to  $240 \times 1216$  for training, validation and testing phases.

### 4.2 IMPLEMENTATION DETAILS

We implement our model in PyTorch Paszke et al. (2019) on 4 NVIDIA 3090 GPUs, using AdamW Loshchilov & Hutter (2017) as optimizer with an initial learning rate of 0.001,  $\beta_1 = 0.9$ ,  $\beta_2 = 0.999$ , weight decay of 0.01. The batch size per GPU is set to 3 and 12 on KITTI DC and NYUv2 datasets, respectively. On the NYUv2 dataset, we train the model for 72 epochs and decay the learning rate by a factor of 0.5 at epochs 36, 48, 60, 72. For the KITTI DC dataset, the model is trained with 100 epochs, and we reduce the learning rate by half at epochs 50, 60, 70, 80, 90. The Appendix outlines more details about network parameters.

Backbone Type	Attention Decoder	Refinement	RMSE↓ (mm)	MAE↓ (mm)	Params.↓ (M)	FLOPs ↓ (G)	
(A)	ResNet34	✗	NLSPN	92.3	36.1	26.4	542.2
(B)	ResNet34	✓	NLSPN	91.4	35.5	28.1	582.1
(C)	Swin-Tiny	✓	NLSPN	92.6	36.4	38.1	634.8
(D)	PVT-Large	✓	NLSPN	90.7	35.2	66.0	410.3
(E)	MPViT-Base	✓	NLSPN	91.0	35.5	83.1	1259.3
(F)	Ours-Tiny	✓	NLSPN	90.9	35.3	45.8	389.4
(G)	Ours-Small	✓	NLSPN	<b>90.0</b>	35.0	82.6	429.6
(H)	Ours-Base	✓	NLSPN	90.1	35.1	146.7	499.6
(I)	ResNet34	✓	✗	106.5	57.8	28.1	581.5
(J)	Ours-Small	✓	✗	100.9	51.9	82.6	429.0
(K)	Ours-Small	✓	CSPN++	90.3	<b>34.9</b>	82.7	446.4
(L)	Ours-Small w/ dual-branch encoders			94.0	36.4	161.0	661.4
(M)	Ours-Small w/o CNN branch			90.7	35.3	29.2	<b>374.6</b>
(N)	Ours-Small w/o Transformer branch			90.6	35.4	59.9	407.7

Scanning Lines	Method	RMSE↓ (mm)	MAE↓ (mm)	iRMSE↓ (1/km)	iMAE↓ (1/km)
1	NLSPN	3507.7	1849.1	13.8	8.9
	DySPN	3625.5	1924.7	13.8	8.9
	Ours-ViT	3507.2	1807.7	12.1	7.8
	Ours	<b>3250.2</b>	<b>1582.6</b>	<b>10.4</b>	<b>6.6</b>
4	NLSPN	2293.1	831.3	7.0	3.4
	DySPN	2285.8	834.3	6.3	3.2
	Ours-ViT	2241.2	795.9	5.8	2.9
	Ours	<b>2150.0</b>	<b>740.1</b>	<b>5.4</b>	<b>2.6</b>
16	NLSPN	1288.9	377.2	3.4	1.4
	DySPN	1274.8	366.4	3.2	1.3
	Ours-ViT	1268.9	360.7	3.3	1.3
	Ours	<b>1218.6</b>	<b>337.4</b>	<b>3.0</b>	<b>1.2</b>
64	NLSPN	889.4	238.8	2.6	1.0
	DySPN	878.5	228.6	<b>2.5</b>	1.0
	Ours-ViT	872.0	226.2	<b>2.5</b>	1.0
	Ours	<b>848.7</b>	<b>215.9</b>	<b>2.5</b>	<b>0.9</b>

(a) Ablation study on network architecture.

(b) Ablation study on scanning lines of LiDAR sensor.

Table 2: **Ablation study on (a) NYU Depth v2 Silberman et al. (2012) and (b) KITTI DC Uhrig et al. (2017) datasets.** In (a), we ablate the settings of our network in the following aspects: the backbone type (including ResNet34 He et al. (2016), Swin-Tiny Liu et al. (2021b), PVT-Large Wang et al. (2021), and MPViT-Base Lee et al. (2022)), the convolutional attention mechanism in decoder and SPN Refinement module. FLOPs are measured with input resolution  $480 \times 640$ . In (b), Ours-ViT denotes that only the Transformer layer is enabled in our proposed block.

### 4.3 EVALUATION METRICS

Following the KITTI benchmark and existing depth completion methods Park et al. (2020); Zhu et al. (2022), given the prediction  $\hat{D}$  and ground truth  $D^{gt}$ , we use the standard metrics for evaluation: (1) root mean square error (RMSE); (2) mean absolute error (MAE); (3) root mean squared error of the inverse depth (iRMSE); (4) mean absolute error of the inverse depth (iMAE); (5) mean absolute relative error (REL).

### 4.4 ABLATION STUDIES AND ANALYSES

We assess the impact of the main components of our CompletionFormer on NYUv2 dataset Silberman et al. (2012). Following previous methods Lin et al. (2022); Park et al. (2020), we randomly sample 500 depth pixels from the ground truth depth map and input them along with the corresponding RGB image for network training. As reported in Tab. 2(a), in RMSE, our novel CompletionFormer in small scale (G) outperforms pure CNNs based method (B) and those pure Transformer-based variants (C, D) counting comparable FLOPs with respect to our model. In particular, compared to the recent MPViT-Base Lee et al. (2022) model (E) which also integrates CNNs and Transformer for feature extraction, our network is not limited to higher accuracy, but also to much lower computational overhead (429.6G FLOPs, while MPViT-Base counts 1259.3G FLOPs). Compared to our baseline (A), *i.e.* NLSPN Park et al. (2020), introducing spatial and channel attention for feature fusion in the decoder (B) further improves the results, with negligible FLOPs increase. Moreover, our models in various scales (E, G, H), benefiting from local and global cues, achieve significant improvement compared to the pure CNN-based baseline (A), while counting fewer FLOPs. To trade-off between accuracy and efficiency, we select our model in small scale (G) as our final architecture for the remaining experiments.

Refinement is also crucial for depth completion methods. Here we ablate the influence of refinement on our CompletionFormer. Compared to a pure CNNs-based network (I), our model with CNNs and Transformer is less affected by the disabling of the NLSPN Park et al. (2020) refinement. Even when we replace the non-local refinement in NLSPN with fixed neighbors in CSPN++ Cheng et al. (2020) (K), the accuracy remains almost the same. It indicates that our CompletionFormer can learn good affinity locally and globally, thus helping to eliminate the problem raised by the limited and fixed range of affinity aggregation in CSPN++ Cheng et al. (2020).

Furthermore, as for previous methods Rho et al. (2022); Tang et al. (2020), we test the dual-branch architecture, which encodes the RGB and depth information separately (L). For feature communication between two branches, we include the spatial and channel attention mechanism Woo et al. (2018) at the end of each stage in the encoder. However, the worse results compared to our one-branch design (G) demonstrates that embedding the multimodal information at the early stage is much more effective and efficient. By individually disabling the convolutional attention (M) and

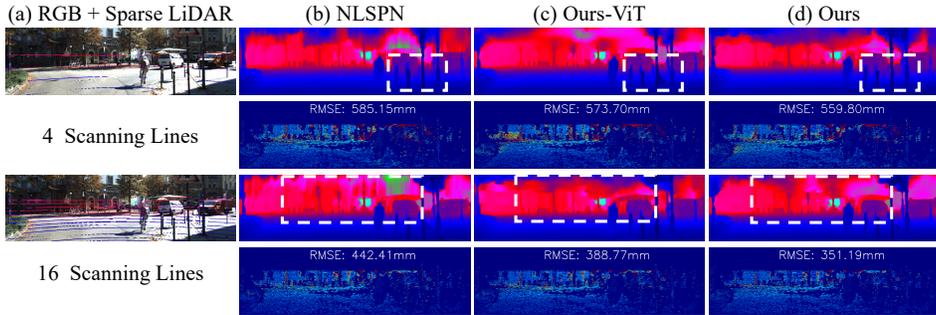


Figure 4: **Qualitative results on KITTI DC selected validation dataset with 4 and 16 LiDAR scanning lines.** We attach the subsampled LiDAR lines to the corresponding RGB image for better visualization. Ours-ViT denotes that only the Transformer layer is enabled in our proposed block. A colder color in depth and error maps denotes a lower value.

Method	RMSE↓ (m)					
	PackNet-SAN	GuideNet	NLSPN	Ours-ViT	Ours	
Sample Number	200	0.155	0.142	0.129	0.130	<b>0.127</b>
	500	0.120	0.101	0.092	0.091	<b>0.090</b>

Table 3: **Sparsity Studies on the NYUv2 Dataset.** Evaluation with 200 and 500 samples.

Transformer layers (N), we witness performance drops in RMSE and MAE. It confirms that local and global contexts are vital for depth completion.

#### 4.5 SPARSITY LEVEL STUDIES AND ANALYSES

Depth maps captured by sensors such as Microsoft Kinect Microsoft and Velodyne LiDAR sensor are unevenly distributed and often contain outliers. To compare the effectiveness of our CompletionFormer with current state-of-the-art methods Guizilini et al. (2021); Tang et al. (2020); Park et al. (2020); Lin et al. (2022) when dealing with this challenging input depth, we manually generate sparse data at different settings for network training and testing. As for noise, since captured depth maps are often corrupted by sensor noise or displacement between RGB camera and depth sensor Qiu et al. (2019), we do not add extra noise to the input depth for experiments.

**Outdoor Scene.** We conduct exhaustive experiments on the KITTI DC dataset Uhrig et al. (2017) to illustrate the robustness of our CompletionFormer when input with different sparsity levels. For all experiments, we train on randomly selected 10 000 RGB and LiDAR pairs from official training data (due to resource constraints) and test on the selected validation dataset provided by KITTI. Furthermore, following Imran et al. (2021), we sub-sample the raw LiDAR points captured by Velodyne HDL-64e in azimuth-elevation space into 1, 4, 16 and 64 lines to simulate the LiDAR-like patterns. All methods have been fully **retrained** using the authors’ code with variable lines of LiDAR inputs (For DySPN Lin et al. (2022), as no public code available, the results have been provided to us by the author using the same selected training list). To get a throughout understanding of the local details encoded by convolution and global context gathered by Transformer, we present the results estimated by pure CNN-based methods (*i.e.* NLSPN and DySPN), a fully Transformer-based variant of our network (Ours-ViT) and our CompletionFormer complete architecture (Ours), which integrates both paradigms. As reported in Tab. 2(b), thanks to the global correlations built by Transformers, our model relying on these blocks only (Ours-ViT) exhibits better results in all metrics as the LiDAR points get sparser. However, solely using Transformer layers makes it difficult to distinguish the objects from the background, as shown in Fig. 4. By coupling the local features and global representations, our complete model (Ours) significantly decreases the errors in all metrics.

**Indoor Scene.** On the NYUv2 dataset Silberman et al. (2012), following Tang et al. (2020), we randomly sample 200 and 500 points from the ground truth depth map to mimic different depth sparsity levels while keeping the ground truth depth used for supervision unchanged. Both our model and NLSPN with publicly available code are **retrained** for a fair comparison, while the results of

Method	KITTI DC				NYUv2		Publication
	MAE↓ (mm)	iMAE↓ (1/km)	iRMSE↓ (1/km)	RMSE↓ (mm)	RMSE↓ (m)	REL↓	
CSPN Cheng et al. (2018)	279.46	1.15	2.93	1019.64	0.117	0.016	ECCV18
DeepLiDAR Qiu et al. (2019)	226.50	1.15	2.56	758.38	0.115	0.022	CVPR19
GuideNet Tang et al. (2020)	218.83	0.99	2.25	736.24	0.101	0.015	TIP20
NLSPN Park et al. (2020)	199.59	0.84	1.99	741.68	0.092	<b>0.012</b>	ECCV20
PENet Hu et al. (2021)	210.55	0.94	2.17	730.08	-	-	ICRA21
ACMNet Zhao et al. (2021)	206.09	0.90	2.08	744.91	0.105	0.015	TIP21
TWIS Imran et al. (2021)	195.58	0.82	2.08	840.20	0.097	0.013	CVPR21
RigNet Yan et al. (2022)	203.25	0.90	2.08	712.66	<b>0.090</b>	0.013	ECCV22
GuideFormer Rho et al. (2022)	207.76	0.97	2.14	721.48	-	-	CVPR22
DySPN Lin et al. (2022)	192.71	0.82	<b>1.88</b>	<b>709.12</b>	<b>0.090</b>	<b>0.012</b>	AAAI22
Ours	<b>183.88</b>	<b>0.80</b>	1.89	764.87	<b>0.090</b>	<b>0.012</b>	-

Table 4: Quantitative evaluation on KITTI DC and NYUv2 Datasets.

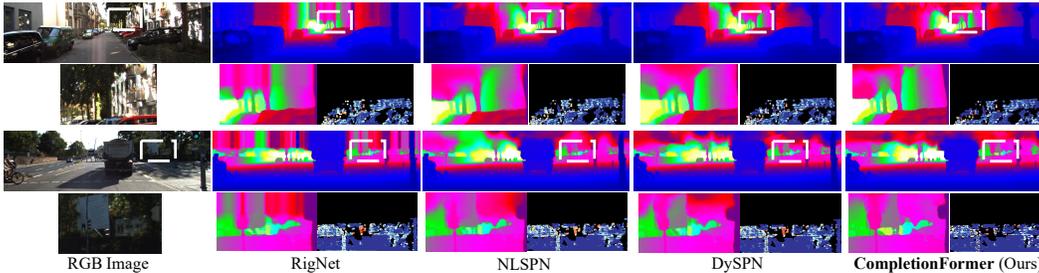


Figure 5: **Qualitative results on the KITTI depth completion test set.** Comparisons of our method against state-of-the-art methods including RigNet Yan et al. (2022), NLSPN Park et al. (2020), DySPN Lin et al. (2022) are presented. We provide RGB images, dense predictions, zoom-in views of challenging areas and corresponding error maps for better visualization.

GuideNet Tang et al. (2020) and PackNet-SAN Guizilini et al. (2021) are taken from the original papers. In Tab. 3, CompletionFormer with both CNNs and Transformers consistently outperforms all other methods in any cases. Qualitative results are provided as [Appendix](#).

#### 4.6 COMPARISON WITH SOTA METHODS

This section comprehensively assesses the performance of state-of-the-art (SOTA) methods. On **indoor**, *i.e.* NYUv2 dataset Silberman et al. (2012), CompletionFormer achieves the best results as reported in Tab. 4. When moving to **outdoor** dataset, we empirically find that our model trained with only  $L_1$  loss achieves the best results on two among four metrics on KITTI depth completion (DC) dataset Uhrig et al. (2017). Specifically, by the time of submission, our network ranks first on the KITTI benchmark on iMAE and MAE metrics and achieves the second-best result on the iRMSE metric among all methods listed in Tab. 4. Qualitative results on the KITTI DC test dataset are provided in Fig. 5. By integrating convolutions and Transformers, our model performs better near depth missing areas (*e.g.* the zoom-in visualization on the second and fourth rows), textureless objects (*e.g.* the cars on the first row) and small objects (*e.g.* the pillars and tree stem far in the distance, on second and fourth rows).

## 5 CONCLUSIONS

This paper proposed a single-branch depth completion network, CompletionFormer, seamlessly integrating convolutions and Transformers into one block. Extensive ablation studies demonstrate the effectiveness and efficiency of our model in depth completion when the input is sparse. This novel design yields state-of-the-art results on indoor and outdoor datasets.

## REFERENCES

- Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.
- Xinjing Cheng, Peng Wang, and Ruigang Yang. Depth estimation via affinity learned with convolutional spatial propagation network. In *ECCV*, pp. 103–119, 2018.
- Xinjing Cheng, Peng Wang, Chenye Guan, and Ruigang Yang. Cspn++: Learning context and resource aware convolutional propagation networks for depth completion. In *AAAI*, volume 34, pp. 10615–10622, 2020.
- Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *ArXiv preprint arXiv:2010.11929*, 2020.
- Vitor Guizilini, Rares Ambrus, Wolfram Burgard, and Adrien Gaidon. Sparse auxiliary networks for unified monocular depth prediction and completion. In *CVPR*, pp. 11078–11088, 2021.
- Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *ECCV*, pp. 630–645. Springer, 2016.
- Mu Hu, Shuling Wang, Bin Li, Shiyu Ning, Li Fan, and Xiaojin Gong. Penet: Towards precise and efficient image guided depth completion. In *ICRA*, pp. 13656–13662. IEEE, 2021.
- Saif Imran, Xiaoming Liu, and Daniel Morris. Depth completion with twin surface extrapolation at occlusion boundaries. In *CVPR*, pp. 2583–2592, 2021.
- Shihao Jiang, Dylan Campbell, Yao Lu, Hongdong Li, and Richard Hartley. Learning to estimate hidden motions with global motion aggregation. In *ICCV*, pp. 9772–9781, 2021.
- Leonid Keselman, John Iselin Woodfill, Anders Grunnet-Jepsen, and Achintya Bhowmik. Intel realsense stereoscopic depth cameras. In *CVPR Workshops*, pp. 1–10, 2017.
- Youngwan Lee, Jonghee Kim, Jeffrey Willette, and Sung Ju Hwang. Mpvit: Multi-path vision transformer for dense prediction. In *CVPR*, 2022.
- Jiankun Li, Peisen Wang, Pengfei Xiong, Tao Cai, Ziwei Yan, Lei Yang, Jiangyu Liu, Haoqiang Fan, and Shuaicheng Liu. Practical stereo matching via cascaded recurrent network with adaptive correlation. *ArXiv preprint arXiv:2203.11483*, 2022a.
- Zhaoshuo Li, Xingtong Liu, Nathan Drenkow, Andy Ding, Francis X Creighton, Russell H Taylor, and Mathias Unberath. Revisiting stereo depth estimation from a sequence-to-sequence perspective with transformers. In *ICCV*, pp. 6197–6206, 2021.
- Zhenyu Li, Zehui Chen, Xianming Liu, and Junjun Jiang. Depthformer: Exploiting long-range correlation and local information for accurate monocular depth estimation. *ArXiv preprint arXiv:2203.14211*, 2022b.
- Yuankai Lin, Tao Cheng, Qi Zhong, Wending Zhou, and Hua Yang. Dynamic spatial propagation network for depth completion. In *AAAI*, 2022.
- Lina Liu, Xibin Song, Xiaoyang Lyu, Junwei Diao, Mengmeng Wang, Yong Liu, and Liangjun Zhang. Fcfr-net: Feature fusion based coarse- to-fine residual learning for depth completion. In *AAAI*, volume 35, pp. 2136–2144, 2021a.
- Sifei Liu, Shalini De Mello, Jinwei Gu, Guangyu Zhong, Ming-Hsuan Yang, and Jan Kautz. Learning affinity via spatial propagation networks. *NeurIPS*, 2017.
- Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, pp. 10012–10022, 2021b.

- Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.
- Fangchang Ma and Sertac Karaman. Sparse-to-dense: Depth prediction from sparse depth samples and a single image. In *ICRA*, 2018.
- Fangchang Ma, Guilherme Venturéli Cavalheiro, and Sertac Karaman. Self-supervised sparse-to-dense: Self-supervised depth completion from lidar and monocular camera. In *ICRA*, pp. 3288–3295. IEEE, 2019.
- Microsoft. Kinect for windows. <https://developer.microsoft.com/en-us/windows/kinect/>.
- Danish Nazir, Marcus Liwicki, Didier Stricker, and Muhammad Zeshan Afzal. Semattnet: Towards attention-based semantic aware guided depth completion. *arXiv preprint arXiv:2204.13635*, 2022.
- Jinsun Park, Kyungdon Joo, Zhe Hu, Chi-Kuei Liu, and In So Kweon. Non-local spatial propagation network for depth completion. In *ECCV*, pp. 120–136. Springer, 2020.
- Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. pp. 8024–8035. Curran Associates, Inc., 2019. URL <http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf>.
- Zhiliang Peng, Wei Huang, Shanzhi Gu, Lingxi Xie, Yaowei Wang, Jianbin Jiao, and Qixiang Ye. Conformer: Local features coupling global representations for visual recognition. In *ICCV*, pp. 367–376, October 2021.
- Jiaxiong Qiu, Zhaopeng Cui, Yinda Zhang, Xingdi Zhang, Shuaicheng Liu, Bing Zeng, and Marc Pollefeys. Deeplidar: Deep surface normal guided depth prediction for outdoor scene from sparse lidar data and single color image. In *CVPR*, pp. 3313–3322, 2019.
- René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. *ArXiv preprint*, 2021.
- Kyeongha Rho, Jinsung Ha, and Youngjung Kim. Guideformer: Transformers for image guided depth completion. In *CVPR*, pp. 6250–6259, 2022.
- Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, pp. 746–760. Springer, 2012.
- Xiuchao Sui, Shaohua Li, Xue Geng, Yan Wu, Xinxing Xu, Yong Liu, Rick Siow Mong Goh, and Hongyuan Zhu. Craft: Cross-attentional flow transformers for robust optical flow. In *CVPR*, 2022.
- Jie Tang, Fei-Peng Tian, Wei Feng, Jian Li, and Ping Tan. Learning guided convolutional network for depth completion. *IEEE TIP*, pp. 1116–1129, 2020.
- Jonas Uhrig, Nick Schneider, Lukas Schneider, Uwe Franke, Thomas Brox, and Andreas Geiger. Sparsity invariant cnns. In *3DV*, pp. 11–20. IEEE, 2017.
- Wouter Van Gansbeke, Davy Neven, Bert De Brabandere, and Luc Van Gool. Sparse and noisy lidar completion with rgb guidance and uncertainty. In *MVA*, pp. 1–6. IEEE, 2019.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *NeurIPS*, 2017.
- Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *ICCV*, pp. 568–578, 2021.

- Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. In *ECCV*, pp. 3–19, 2018.
- Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *NeurIPS*, 34, 2021.
- Xin Xiong, Haipeng Xiong, Ke Xian, Chen Zhao, Zhiguo Cao, and Xin Li. Sparse-to-dense depth completion revisited: Sampling strategy and graph construction. In *ECCV*, pp. 682–699. Springer, 2020.
- Weijian Xu, Yifan Xu, Tyler Chang, and Zhuowen Tu. Co-scale conv-attentional image transformers. In *ICCV*, pp. 9981–9990, 2021.
- Zheyuan Xu, Hongche Yin, and Jian Yao. Deformable spatial propagation networks for depth completion. In *ICIP*, pp. 913–917. IEEE, 2020.
- Zhiqiang Yan, Kun Wang, Xiang Li, Zhenyu Zhang, Baobei Xu, Jun Li, and Jian Yang. Rignet: Repetitive image guided network for depth completion. *ECCV*, 2022.
- Yinda Zhang and Thomas Funkhouser. Deep depth completion of a single rgb-d image. In *CVPR*, pp. 175–185, 2018.
- Yongchi Zhang, Ping Wei, Huan Li, and Nanning Zheng. Multiscale adaptation fusion networks for depth completion. In *IJCNN*, pp. 1–7, 2020.
- Chaoqiang Zhao, Youmin Zhang, Matteo Poggi, Fabio Tosi, Xianda Guo, Zheng Zhu, Guan Huang, Yang Tang, and Stefano Mattoccia. Monovit: Self-supervised monocular depth estimation with a vision transformer. In *3DV*, 2022.
- Shanshan Zhao, Mingming Gong, Huan Fu, and Dacheng Tao. Adaptive context-aware multi-modal network for depth completion. *TIP*, pp. 5264–5276, 2021.
- Yiqi Zhong, Cho-Ying Wu, Suya You, and Ulrich Neumann. Deep rgb-d canonical correlation analysis for sparse depth completion. *NeurIPS*, 2019.
- Yufan Zhu, Weisheng Dong, Leida Li, Jinjian Wu, Xin Li, and Guangming Shi. Robust depth completion with uncertainty-driven loss functions. In *AAAI*, 2022.

## A APPENDIX

### A.1 QUALITATIVE RESULTS ON NYUV2 DATASET

Qualitative results concerning the NYUv2 dataset Silberman et al. (2012) are provided in Fig. 6. In both visualized cases, we can notice the improved results yielded by our CompletionFormer compared to NLSPN Park et al. (2020). Especially for the transparent regions near the windows in both cases, with local details of convolution and global cues of Transformer, our complete model (Ours) predicts clear object boundaries while NLSPN and Ours-ViT give blurry estimations.

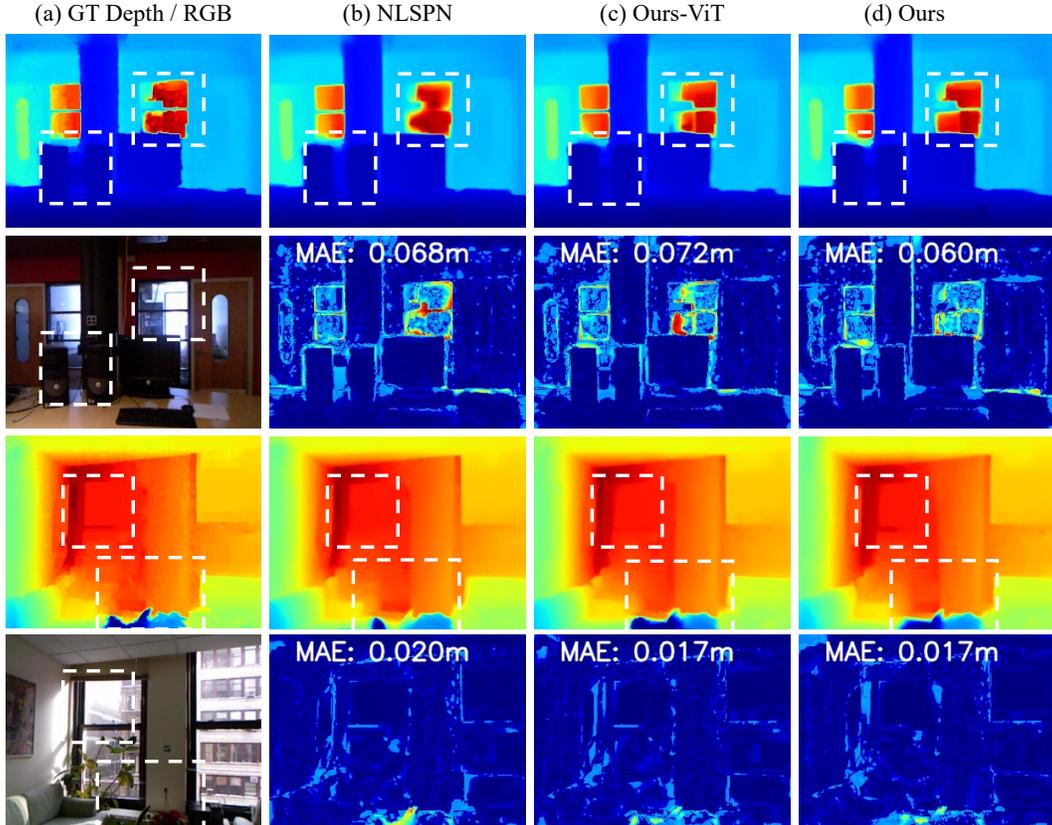


Figure 6: **Qualitative results on NYUv2 dataset.** Comparisons of our method against state-of-the-art method, *i.e.* NLSPN Park et al. (2020) are presented. We provide RGB images, and dense predictions. The colder the colors of the error map, the lower the errors. Ours-ViT denotes that only the Transformer layer is enabled in our proposed block.

### A.2 MODEL ARCHITECTURE DETAILS

To better understanding our architecture and to ease reproducibility, we present the network parameters of our CompletionFormer in Tab. 5. We will make **code available** in case of acceptance.

Name	Layer setting	Output dimension
<b>RGB and Depth Embedding</b>		
input		RGB Image: $H \times W \times 3$ Sparse Depth: $H \times W \times 1$
conv_separate	Conv $3 \times 3, 48$ for RGB Image Conv $3 \times 3, 16$ for Sparse Depth	RGB Feature: $H \times W \times 48$ Sparse Depth Feature: $H \times W \times 16$
conv1	concat [RGB, Sparse Depth Feature] Conv $3 \times 3, 64$	$H \times W \times 64$
<b>Joint Convolutional Attention and Transformer Encoder</b>		
conv2	ResNet34 He et al. (2016) BasicBlock $\times 3$	$H \times W \times 64$
conv3	ResNet34 He et al. (2016) BasicBlock $\times 4$	$\frac{1}{2}H \times \frac{1}{2}W \times 128$
conv4	Joint Convolutional Attention and Transformer Block $\times 3$	$\frac{1}{4}H \times \frac{1}{4}W \times 64$
conv5	Joint Convolutional Attention and Transformer Block $\times 3$	$\frac{1}{8}H \times \frac{1}{8}W \times 128$
conv6	Joint Convolutional Attention and Transformer Block $\times 6$	$\frac{1}{16}H \times \frac{1}{16}W \times 320$
conv7	Joint Convolutional Attention and Transformer Block $\times 3$	$\frac{1}{32}H \times \frac{1}{32}W \times 512$
<b>Decoder</b>		
dec6	ConvTranspose $3 \times 3$ , stride = 2, 256 Convolutional Attention Layer	$\frac{1}{16}H \times \frac{1}{16}W \times 256$
dec5	concat [dec6, conv6] ConvTranspose $3 \times 3$ , stride = 2, 128 Convolutional Attention Layer	$\frac{1}{8}H \times \frac{1}{8}W \times 128$
dec4	concat [dec5, conv5] ConvTranspose $3 \times 3$ , stride = 2, 64 Convolutional Attention Layer	$\frac{1}{4}H \times \frac{1}{4}W \times 64$
dec3	concat [dec4, conv4] ConvTranspose $3 \times 3$ , stride = 2, 64 Convolutional Attention Layer	$\frac{1}{2}H \times \frac{1}{2}W \times 64$
dec2	concat [dec3, conv3] ConvTranspose $3 \times 3$ , stride = 2, 64 Convolutional Attention Layer	$H \times W \times 64$
<b>Initial Depth, Confidence, Non-local Neighbors, Affinity Prediction Head</b>		
dec1	concat [dec2, conv2] Conv $3 \times 3, 64$	$H \times W \times 64$
dec0	concat [dec1, conv1] Conv $3 \times 3, \eta$	$H \times W \times \eta$
<b>SPN Refinement</b>		
refine	Spatial Propagation Network Park et al. (2020) with recurrent time $K = 6$	$H \times W \times 1$

Table 5: Network parameters of CompletionFormer. ‘concat’ means performing concatenate operation at Channel dimension. For each prediction head, it takes almost the same design, and only the output channel  $\eta$  is dependent on the output type, *e.g.*  $\eta = 1$  for initial depth prediction.